

# Inktober 2017

Guillaume Pelletier-Auger

26 septembre 2017

## Résumé

Diverses rencontres entre l'encre et les algorithmes.

## 1 Introduction

## 2 Les éléments qu'il me faut

J'ai besoin de trois modes : *drawGraph*, *detectGraph* et *animate*.

Et j'ai besoin d'un ensemble de *systems* entre lesquels je peux alterner à ma guise.

## 3 Définition des systèmes

Les systèmes sont définis par un ensemble de graphes. Les graphes ont des sommets. Ces sommets ont des comportements (*behaviours*). La seule chose que fait l'instance *geo* de p5, c'est d'afficher chacun des sommets de chaque graphes selon sa couleur.

## 4 Les grenouilles

L'objet *Grenouille* est à mi-chemin entre une tortue de Papert et une particule classique. Elle est définie par un ensemble d'instructions écrites en *turtle talk*, et cet ensemble est parcouru en boucle. À chaque pas, la nouvelle position de la grenouille est calculée en considérant à la fois les forces extérieures qui l'affectent et cet ensemble d'instructions qui lui est intrinsèque.

L'objet *Grenouille* doit donc avoir une donnée *currentStep* qui est incrémentée à chaque pas puis remis à zéro lorsque toutes les instructions ont été parcourues.

Et à chaque *step*, la position que devrait gagner la grenouille à la fin du pas, cette position à laquelle la grenouille espère accéder, cette position de rêve est considéré comme un attracteur très fort.

Le *dialecte de la grenouille* doit ainsi différer du *turtle talk* : un changement d'angle, que ce soit LEFT ou RIGHT, n'est pas considéré comme un pas. Un pas consiste plutôt à une paire de nombres : l'angle de rotation et la longueur du pas. Chacun de ces deux nombres peut être zéro, mais il doit exister.

Ah, mais en fait, je pourrais parler à ma grenouille en *turtle talk*, et ensuite un programme pourrait convertir le tout en *dialecte de grenouille*.